On the Usability of Authenticity Checks for Hardware Security Tokens

Katharina Pfeffer
SBA Research

Alexandra Mai
SBA Research

Adrian Dabrowski
University of California, Irvine

Matthias Gusenbauer
Tokyo Institute of Technology & SBA Research

Philipp Schindler
SBA Research

Edgar Weippl
University of Vienna, Austria

Michael Franz
University of California, Irvine

Katharina Krombholz
CISPA Helmholtz Center for Information Security

Abstract

The final responsibility to verify whether a newly purchased hardware security token (HST) is authentic and unmodified lies with the end user. However, recently reported attacks on such tokens suggest that users cannot take the security guarantees of their HSTs for granted, even despite widely deployed authenticity checks. We present the first comprehensive market review evaluating the effectiveness and usability of authenticity checks for the most commonly used HSTs. Furthermore, we conducted a survey ($n = 194$) to examine users' perceptions and usage of these checks.

We found that due to a lack of transparency and information, users often do not carry out—or even are not aware of—essential checks but rely on less meaningful methods. Moreover, our results confirm that currently deployed authenticity checks suffer from improperly perceived effectiveness and cannot mitigate all variants of distribution attacks. Furthermore, some authenticity concepts of different manufacturers contradict each other. In order to address these challenges, we suggest (i) a combination of conventional and novel authenticity checks, and (ii) a user-centered, transparent design.

1 Introduction

Due to an abundance of reported malware and CPU vulnerabilities [32, 46, 76, 85, 117], the establishment of trust has in recent years shifted from general-purpose computers to specialized single-application devices, i.e., hardware security tokens (HSTs). HSTs (e.g., Two-Factor Authentication (2FA) tokens or cryptocurrency hardware wallets) promise to keep the stored secrets secure, even if attackers control the client computer. Consequently, these tokens have experienced an enormous market growth during the last decade [72, 73]; all major browsers and many large service providers now support 2FA tokens [110]. Similarily, in the cryptocurrency ecosystem hardware wallets are considered the most secure way to manage keys and sign transactions.

However, known attacks using modified, replaced, or counterfeit tokens [60, 75, 86, 95, 109] raise the questions whether this shift of trust is justified and how users may verify the authenticity of their HSTs. In the context of this paper, authenticity checks are defined as (i) conventional attestation[1] methods, and (ii) haptic and visual inspection of the packaging, casings and electronics.

Despite extensive research focusing on authenticity checks for computing devices [6, 27, 63, 70, 80, 99], little attention has been paid to whether and how these checks can be applied to HSTs. For HSTs, no categorization of authenticity checks concerning their effectiveness, efficiency, or usability exists. As a result, HST manufacturers[2] have no directives or best practices available for designing and implementing defenses. Although end users play a central role in judging a token's authenticity, no human-centered research has so far been pursued in this area. It remains therefore unclear how users can make sure that a token is genuine and/or has not been manipulated. In particular, the following research questions arise:

(RQ1) How effective are currently deployed authenticity checks of HSTs in defending against possible attacks?

(RQ2) How do users perceive and use the provided authenticity checks?

(RQ3) Which (combination of) authenticity checks can maximize security and usability?

To answer these questions, we contribute:

- A **market review** of authenticity checks deployed in HSTs, yielding an evaluation framework for comparing their effectiveness and usability.
- A **quantitative survey** ($n = 194$) to understand users' perception, awareness and usage of the investigated authenticity checks as well as related trust decisions.
- **Actionable recommendations** pointing out directions for the best (combination of) authenticity checks.

In this paper, we assessed (i) cryptocurrency hardware wallets, as they are high-value targets, and (ii) Universal Second

[1] Attestation proves that no unwarranted modifications to the software or hardware took place [13].

[2] Companies that (mostly) perform the final assembly as well as the development, design, and advertisement of a product.

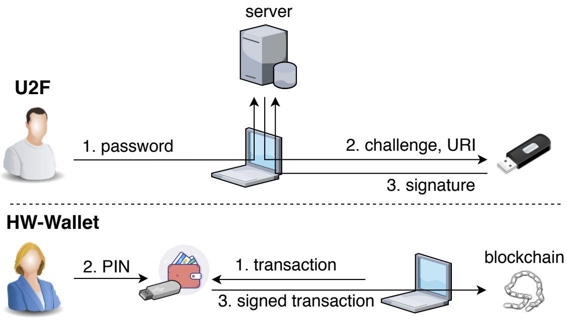

Figure 1: Simplified U2F/HW-Wallet Authentication Models

Factor (U2F) tokens which are widely used for 2FA. We focus on attacks which take place during the distribution process, consequently defining them as *distribution attacks*. Many of these attacks can be carried out with no technical expertise, in reasonable time, and with low financial cost.

2 Hardware Security Tokens (HSTs)

The main purpose of HSTs is to securely store cryptographic keys. Depending on the application context, the tokens can carry out different operations such as proving a user's identity or signing a cryptocurrency transaction.

2.1 Hardware Wallets

Hardware wallets are used to store the user's private key in tamper-resistant storage and to sign transactions. For this purpose they communicate with a PC via USB (or alternatively, NFC or Bluetooth) as shown in Figure 1. A dedicated client software constructs transactions and sends them to the hardware device for signing. The device signs after explicit and successful user approval (e.g., PIN or hash comparison). When used for the first time, the device generates a seed for deterministic private keys which never leaves the wallet [33].

Hardware wallets fully control the secrets for signing. They have access to the complete transactions including sender and receiver address, an optional change address, and the payment amount. The increased usage of hardware wallets makes them a valuable attack target.

2.2 U2F Tokens

Currently, the most popular U2F tokens are the YubiKeys (cf. the corresponding media and industry attention [37, 50] and Google trend analysis [34]). YubiKeys provide multi-factor and password-less authentication for logins. They currently support, amongst others, the following protocols: (i) Universal 2nd Factor (U2F/FIDO2), (ii) one-time password (OTP), (iii) Smart card, and (iv) PGP. In this paper, we focus on YubiKey's U2F functionality.

The tokens are shipped with a pre-configured public/private identity and an AES key which serves as the master secret for deriving subsequent authentication keys. Users may also generate their own keys. The initialized YubiKey communicates with the computer via USB [26]—including mimicking a keyboard [114]—or NFC.

In U2F mode, the user sends the password to the server which replies with a challenge (see Figure 1); the user's presence is verified by touching the YubiKey sensor. Then the YubiKey utilizes a private key generated per-service to calculate a response to the server's challenge, i.e., a signature. The browser never learns the private key and the YubiKey never sees the user's password, hence there is no single point of trust. An application ID derived from the URI is included in the signature to prevent phishing attacks.

3 Related Work

The **usability challenges** of HSTs and 2FA schemes have been extensively studied. Bonneau et al. [10] showed that most password-less web authentication methods, including hardware tokens, outperform passwords regarding security but are weaker concerning usability. Payne et al. [81] explored user perceptions of the Pico authentication token. They found that tokens increase the user's responsibility to mitigate security risks, which is usually perceived as inconvenient.

More recently, Acemyan et al. [2] found severe usability issues in Google 2FA features. Studies by Reynolds et al. [87, 88] revealed usability issues in the set-up and usage of U2F tokens in enterprise and non-enterprise settings. Das et al. [18] conducted a study with YubiKey users, reporting usability and trust issues as well as misconceptions about the token's benefits. In two other user studies, Ciolino et al. [14] confirmed uncertainties about the security benefits of 2FA tokens and identified usability issues of online services secured with 2FA. Human-centered research in the domain of hardware wallets mainly focused on the fact that humans usually fail to manually compare long hashes [20, 42, 104], which is required by most devices. So far, user perceptions of HST authenticity and related decisions regarding trust—as presented in this paper—have not been examined.

In order to prevent and detect supply-chain tampering of software and hardware [7, 17], various **attestation approaches** have been suggested. *Software attestation* [6, 63, 99, 101] aims at validating the authenticity of code by verifying software modules (e.g., calculating a hash or MAC). *Hardware attestation* aims to ensure the authenticity of hardware components. Approaches range from (i) dedicated hardware designs (e.g., tamper-proof environments for isolation of security-critical functionality [66], single-piece or openable enclosures [36], tamper-evident seals [48]) to (ii) sensors that detect suspicious behavior [31, 39, 62] to (iii) hardware metering [52, 53] (e.g., using PUFs [38, 44, 91] or IC fingerprinting [3]).

However, each of these approaches poses different challenges [5]. Hence, solutions must be found that combine several methods and are tailored to each use case and threat model. We discuss which of these approaches are currently implemented by popular HSTs and evaluate how effective they are against real-world attack vectors (see Section 4.1). Dauterman et al. [19] introduced a two-party key and signature generation protocol as a (partial) solution to defend against faulty or backdoored tokens. We discuss their scheme in Section 7.4. While previous research mainly focused on a theoretic evaluation of individual attestation methods, our work assesses these methods' usability for HSTs operated by average end users.

4 Threat Model

The attackers' aim is to exfiltrate or pre-load secrets stored on the HST (i.e., keys or cryptographic seeds), interrupt its availability, or ask for ransom [12]. Attackers can replace or modify HSTs anywhere and anytime between the token leaving the manufacturer and arriving at the end user. This includes building fraudulent HSTs and selling them directly to end users, inserting them into the re-seller hierarchy, or intercepting and replacing shipments during delivery. An attacker might also buy a genuine token and return a tampered one to the vendor who usually does not check the returned devices before redistribution [43]. We define this set of attacks as *distribution attacks*.

Attacks performed after the initialization of a hardware device such as Man-in-the-middle and phishing attacks are out of this work's scope. We assume token manufacturers and designers to be trustworthy, meaning that they are not altering hardware or firmware. Still, fraudulent manufacturers or parties that (re-)sell counterfeit tokens do exist. We include nation-state attackers if they modify or replace HSTs on their route from the manufacturer to the end user. Finally, for authenticity checks involving the client software we assume that this software is not compromised.

Generally, attacks can be aimed at one or more specific targets (*targeted attacks*) or at multiple unspecific targets (*large-scale attacks*). Targeted attacks concern U2F token users (e.g., campaign teams, activists, journalists, IT administrators) and individual hardware wallet users holding high amounts of cryptocurrencies. In contrast, large-scale attacks mainly affect hardware wallets due to the expected monetary gain. Even though reported attacks on HSTs are still rare, their relevance for HST users is justified given the financial and/or reputational losses. Also, talks and papers on the construction of counterfeit tokens [60, 75] emphasize that HST authenticity should be addressed before a larger number of attacks can take place.

4.1 Attack Vectors

We conducted an extensive review of scientific literature, security conferences, and blog articles to understand the threat landscape of HSTs. We then extracted attack vectors which are categorized in *software*, *hardware*, and *secret extraction*. We define **attack vectors** as ways or means by which attackers can carry out attacks. Examples of **attack scenarios** are described in Section 4.2. A visual mapping of attack vectors and scenarios can be found in Table 1.

Although the attack vectors are the same for hardware wallets and YubiKeys, the actual attack scenarios vary, since YubiKeys and other U2F tokens—unlike hardware wallets—do not present a single point of trust when used for 2FA. Here, the key material alone is useless since the token never learns the user's password. However, if the token is used for single-factor authentication, a counterfeit token that manages to exfiltrate or pre-load secrets can achieve authentication. Consequently, using a U2F token without a supplementary factor increases the probability of severe attacks being successful.

4.1.1 Software

Firmware modifications can be conducted by reverse-engineering code, changing open-source software, or taking advantage of firmware vulnerabilities. Alternatively, attackers might exploit security risks of the USB interface (**USB exploits**) [106] or **pre-initialize tokens**. Programmers for example can conduct such attacks without any special additional knowledge [86, 98, 108, 109].

4.1.2 Hardware

Attackers can add and wire-up additional components to the token—so-called **hardware implants** such as a GSM module or Bluetooth transceiver. This aims at leaking secrets or remotely controlling the tokens. Alternatively, Integrated Circuit **(IC) modification** is possible to introduce vulnerabilities or backdoors [8].

Lastly, attackers can build **token replicas**, which is feasible for hardware wallets and YubiKeys [60, 75]. Instructions on how to create HST replicas are publicly available and can be implemented without any expert knowledge. If successful, attackers gain full access to the design of the hardware and firmware and may modify them to their own advantage.

4.1.3 Secret Extraction

Hardware and software attacks use various approaches to extract secrets from a token, e.g., keys or cryptographic seeds. Most commonly, information can be derived from **fault injections**, **timing side-channels** (including transient execution attacks (see Section 5.2.3), **IC microprobing**, or **bus snooping** [8]. Some of these attacks require expensive equipment and in-depth knowledge. However, respective instructions are publicly available and prices for the required equipment are falling, thus facilitating secret extraction [43].

4.2 Attack Scenarios

Run-time seed or key exfiltration (in-band): The attacker replaces the HST and/or modifies its firmware so that it leaks secrets through in-protocol covert channels[3] via the signature [23] or other parts of the transaction [11].

Run-time seed or key exfiltration (out-of-band): The attacker modifies or replaces the HST's software or hardware so that it leaks secrets through covert channels outside the

[3] Covert channels [116] intentionally hide the communication between two parties, whereas side channels unintentionally leak internal state.

protocol (e.g., using Bluetooth [60], Wi-Fi, GSM [75], or USB exploits.

Delivery-time seed or key extraction: Through side channels, bus snooping, IC microprobing, or fault injection, the attacker extracts pre-configured keys or seeds that allow key determination [60, 78, 82]. This attack is relevant for YubiKeys which are shipped with manufacturer-chosen seeds—and as long as users do not programm their own secrets later on—but is infeasible for hardware wallets since customers (should) initialize the tokens themselves.

Seed or key fixation: Using hardware implants, token replication, or firmware modification, the attacker pre-loads a key to the token, makes the key computation deterministic, or pre-initializes a hardware wallet and inserts a fake recovery sheet [98].

Predictable RNG modification: The attacker makes the Random Number Generator predictable [68] by using hardware implants, replicated HSTs, or IC/firmware modifications. Alternatively, the attacker exploits unintentionally weak randomness [113].

USB pivoting: The attacker uses the USB interface to infect the computer with malware [106], trigger buffer overflows in the client software, or emulate a keyboard similar to a USB Rubber Ducky [71]. This can be used for attacks on YubiKeys that require leaking information such as usernames or passwords in addition to the key or seed.

Table 1: Evaluation Framework

Effectiveness (market review)
○ no prevention
● strong protection
◐ complicates attack/decreases usefulness

Attack Vector Usage in Scenarios
✔ potentially used

			Hardware implants	Token replication	IC modification	Firmware modification	USB exploit	Token pre-initialization	Timing side-channels	Bus snooping	IC microprobing	Fault injection
			Attack Vectors									
			Hardware			Software			Secret Extraction			
Attestation / Countermeasure	Pack.	Tamper-evident	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐
		Holographic sticker	○	○	○	○	○	○	○	○	○	○
	Hardware	Single-piece cast	●	◐	◐	○	○	○	○	◐	◐	○
		Openable device	◐	◐	○	○	○	○	○	○	○	○
		Secure element (co-processor)	○	●	●	○	○	○	●	◐	●	●
		Secure CPU	◐	●	●	○	○	○	●	●	●	●
	Software	Local firmware attestation	○	○	○	●	○	○	○	○	○	○
		Remote firmware attestation	○	◐	○	●	○	○	○	○	○	○
		Key attestation	○	●	○	◐	○	○	○	○	○	○
		Manual firmware load	○	◐	○	●	○	◐	◐	◐	◐	◐
Attack Scenarios[1]	Key compromise	Runtime seed or key exfilt. (in-band)	–	✔	–	✔	–	–	–	–	–	–
		Runtime seed or key exfilt. (out-of-band)	✔	✔	✔	✔	✔	–	–	–	–	–
		Delivery-time seed or key extraction	–	–	–	–	–	–	✔	✔	✔	✔
		Seed or key fixation	✔	✔	–	✔	–	✔	–	–	–	–
		Predictable RNG modification	✔	✔	✔	✔	–	–	–	–	–	–
	Other	USB pivoting	✔	✔	✔	–	✔	–	–	–	–	–
		Ransom attack	✔	✔	–	✔	–	–	–	–	–	–

[1] For U2F tokens, the attacker needs a second source for retrieving username/password.

Ransom attack: Using hardware implants, token replication, or firmware modification, the attacker modifies the token which then stops operating after some time, demanding a ransom to resume operation or release the key material. This attack is especially efficient with hardware wallets which are in control of the secret key material and the derivation of the used addresses. Consequently, a fraudulent hardware wallet can prevent the user from obtaining a trustworthy backup of the secret material by displaying a wrong recovery seed. Alternatively, it can generate addresses which cannot be derived from a correct backup [12]. This attack is limitedly feasible for YubiKeys, as backup keys are usually generated.

5 Market Review of Authenticity Checks

To answer *RQ1*, we assessed four different models of the U2F YubiKey and the five most recent hardware wallets[4] from the three most popular vendors at the time of writing [73].

We chose YubiKeys as representatives of FIDO/U2F-tokens based on previous research [18, 88] and Yubico's role as the currently leading U2F-manufacturer (see Section 2.2). To ensure that YubiKeys are indeed representative of attestation and packaging methods used in the U2F token industry, we surveyed other U2F-certified tokens (e.g., Google Titan [35], Thetis BLE/FIDO U2F [105], Feitian ePass FIDO [30]) before the market review. We found that YubiKeys' methods are the most comprehensive in the industry (see Appendix, Table 6).

We examined nine widely used tokens and are therefore confident that our results are representative (although not exhaustive) of authenticity checks deployed in today's HSTs.

5.1 Methodology

In order to assess which authenticity checks were deployed and if they were usable, we performed a set of cognitive walkthroughs [83]. A cognitive walkthrough is a technique for expert usability inspection of a system and commonly applied in user-centered security research [28]. Thereby, an expert steps through a set of actions while considering the interface behavior and its effect on the user. For this study, two usable security researchers walked through the actions a user has to perform when receiving an HST and initializing it, including the examination of the packaging. For each of these actions we asked: *Does the user understand what they are supposed to do? Does the user know how to do it? After the action is done, does the user know whether it was successful?* We used the findings of the cognitive walkthroughs to design our quantitative survey (see Section 6).

Additionally, we consulted the manufacturers' documentation to obtain a complete list of the deployed authenticity checks. We then connected these findings with the data of our cognitive walkthroughs and established three categories for authenticity checks (Inter-rater reliability: Krippendorff's

[4] Some hardware wallets offer U2F functionality as an add-on. However, we focused only on their core funcionality.

α=.91): (i) packaging, (ii) hardware, and (iii) software. In order to systematically evaluate each authenticity check, we mapped them to attack vectors (see Section 4.1), thus building an evaluation framework for comparing the effectiveness of current and future authenticity checks (see Table 1).

5.2 Results

In this section we present a comprehensive evaluation of the usability and effectiveness of currently deployed authenticity checks. Table 2 shows the investigated devices and their authenticity checks. Table 1 illustrates our evaluation framework by mapping deployed authenticity checks to attack vectors. The effectiveness assessment is based on the currently deployed best-case implementation of every method, as found in the market review. We also discuss deviations from the best case scenario, since flawed or inadequate implementations make every method ineffective.

5.2.1 Packaging

Trezor One and one of the tested YubiKeys were shipped in a **tamper-evident package**, meaning that it shows if a package has been opened. The hardware wallets arrived in cardboard boxes or shrink-wrap plastic. YubiKey recently switched to hard shells (i.e., tamper-evident blister packaging), but older models were delivered in plastic sleeves. A lot of manufacturers additionally provide pictures of the original packaging on their websites and encourage customers to report and return damaged shipments. Six of the assessed devices came with **holographic stickers**.

Effectiveness: Tamper-evident packages render an attack slightly more difficult: an attacker would have to re-package a modified device in a genuine-looking way. Still, all types of packaging can be reproduced; paper boxes and standard plastic sleeves are easy and cheap, whereas reconstructing tamper-evident plastic wraps is more expensive, since special-purpose machines are needed. Hence, the latter only pays off if attacks are carried out on a large scale. Holographic stickers only provide a low level of protection against distribution attacks. They can be removed with a common blow dryer [75], and new ones are easy to come by [45].

Usability: Some packages are destroyed when opened, making any tampering clearly visible. Also, there are self-destroying holographic stickers which cannot be easily replaced. However, other types of packaging—e.g., simple paper boxes—do not show obvious signs after they have been opened. Only a few manufacturers provide information on what the original package and holographic sticker(s) should look like. Therefore, users often do not have any possiblity to verify if the packaging is the original one.

5.2.2 Hardware (Enclosure)

Manufacturers take two contrary approaches in order to secure the token body. One way is to use a **single-piece cast**. Yubico chose this method and encourages users to check the integrity of tokens through visual inspection. The other way is a token which can easily **be opened to compare the inside to reference pictures** on the manufacturer's website. The latter was the case for two of the assessed hardware wallets.

Table 2: Device and Feature Overview

● fulfilled/implemented/included
◑ sometimes ○ not fullfilled
– not applicable ? undisclosed

		Ledger Nano S	Ledger Blue	Trezor One	Trezor Model T	Keepkey	YubiKey 5	YubiKey 4 Neo	YubiKey 4	Yubico Sec. Key
Pack.	Tamper-evident	○	○	●	○	○	●	◑[1]	◑[1]	◑[1]
	Holographic sticker	○	○	●	●	●	○	◑[1]	◑[1]	◑[1]
Hardware	Single-piece cast	–[2]	–[2]	○	○	○	●	●	●	●
	Openable device	●	●	○	○	○	–[2]	–[2]	–[2]	–[2]
	Secure CPU	○	○	○	○	○	●	○	○	○
	Secure element (co-processor)	●	●	○	○	○	○	●	●	●
Software	Local firmware attestation	●	●	●	●	●	?	?	?	?
	Remote firmware attestation	●	●	○	○	○	○	○	○	○
	Key attestation	●	●	○	○	○	●	●	●	○
	Manual firmware load	○	○	●	●	○	○	○	○	○

[1] Packaging changed multiple times in recent years. [2] Mutually exclusive.

Effectiveness: Some single-piece cast devices are easy to break by using household chemicals [40]. Therefore, the YubiKey 5 series is made with a more chemical-resistant thermoplastic. Since the electronics are tightly molded, it is infeasible for an attacker to add hardware components. Creating token replicas or modifying ICs of a single-piece cast token requires a very elaborate process, since an original-looking cast has to be built from scratch. At the same time, these attacks have the advantage that the built-in hardware does not need to look genuine, only the case does (assuming that end users usually do not x-ray their devices). Lastly, it is feasible—although very elaborate—to conduct bus snooping or IC microprobing by drilling small, resealable holes into the case.

On the one hand, openable tokens and visual inspectability enable users to discover implants and make token replica attacks more difficult. On the other hand, openable tokens give attackers easy access as well. It can be assumed that well-made, subtle hardware implants would not be noticeable to users. IC modifications are also possible, since chips come in standardized packages which are easily re-constructed.

Usability: Visually comparing the interior of the device with manufacturer-provided pictures is a cumbersome and error-prone method. Users might damage the case when opening it, which reduces the feasibility and usability of this approach. Tamper-resistant casts do not exhibit these usability issues.

5.2.3 Hardware (Circuit)

Electronic signals on the printed circuit board (PCB) or within an IC are subject to interception and manipulation. Shielding of critical data and the respective circuitry can be accomplished through a **secure CPU**, or by integrating an external co-processor (**secure element**) on the PCB. The keys reside inside the CPU or element and never leave it. Many software-based authenticity checks only provide strong protection when implemented in such hardened CPU design and architecture (e.g., firmware or key attestation). In our wallet sample, only Ledger facilitates a secure element [59]. Trezor in fact argues against secure elements, as they are closed source software, and postulates that if "*secure elements [are] widely used, it will increasingly attract the attention of hackers*" [94].

Effectiveness: Many secure CPUs and secure elements are designed with the hardware attacker in mind and provide the respective prevention measures. They usually employ side-channel-resistant design and tamper-detection circuits within the IC. However, even if secure CPUs (including enclaves) are used, transient execution attacks [51, 64, 107] can extract secrets via i.a. cache-timing side-channels once the attacker achieves code execution. Such features are still seldom found in low-end microcontrollers as used in wallets and authentication tokens. All wallets in our device overview use ARM Cortex M0-M4 architectures which neither employ data caches nor transient execution. External secure elements are vulnerable to hardware implants [93] and susceptible to bus snooping. Rewiring or snooping signals on a PCB requires far less equipment than doing the same on an IC. Moreover, a fraudulent firmware could, in theory, still leak secrets via a physical channel [79]. Technical challenges of ARM TrustZone have been reported recently [84, 92].

Usability: In this case, the user is not involved. However, if users are aware of these measures, they can accordingly base their trust decisions on them.

5.2.4 Software (Automatic)

For all assessed hardware wallets, the authenticity of the boot loader and/or firmware is checked by a hash or signature verification. This is carried out either by the firmware, the boot loader, or the secure element. The simplest form of software attestation provided by our tested devices is **local firmware validation**. Thereby, the boot loader validates the integrity of the firmware (by conducting a signature check), or vice versa. For two of the assessed devices, the secure element locally attests the authenticity of the micro-controller unit via a signature check. A more sophisticated approach is to use **remote firmware attestation** where the internal status of the device is attested by a trusted third party (e.g., by utilizing challenge-response protocols).

Yubico is very secretive about any of their implemented automatic software attestation methods. Thus, it remains unclear whether such methods are applied to our tested YubiKeys.

Effectiveness: Remote firmware attestation is more effective than local methods since it complicates token replication. With remote attestation in place, attackers would have to mimic the third-party attestation protocol. Generally, the effectiveness of all firmware attestation methods is increased if secure CPUs or secure elements are involved. Despite these approaches being implemented, several attacks on firmware have been carried out. Although manufacturers usually fix these vulnerabilities, their existence—even if only for a short time—poses a threat which is hard to defeat. As the manufacturing of hardware tokens becomes more sophisticated and globally distributed, and the time-to-market constantly shortens, the probability of software vulnerabilities is growing [8]. Furthermore, automatic software attestation methods are ineffective against hardware implants, IC modification, USB exploits, token pre-initialization, and secret extraction.

Usability: Automatic software checks do not require user interaction, hence they do not cause any usability issues. However, if these checks are not visible and/or known to users, they are not able to make related trust decisions.

5.2.5 Software (Manual)

YubiKeys come with a pre-loaded **attestation key** and a manufacturer-signed attestation certificate. Users can manually verify the authenticity of their YubiKey by visiting a sub-page of the manufacturer's website [115]. With regard to our sample, Yubico and Ledger do check the attestation key of the devices. A server (e.g., an online banking service) can optionally request an attestation certificate from a YubiKey during user registration to check the device's authenticity. YubiKeys additionally have the option to run Personal Identity Verification (PIV) attestation for newly generated keys to ensure that a certain asymmetric key was generated on the device and not imported from elsewhere.

A further attack prevention method (used by two of the tested hardware wallets) is to ship tokens without firmware, thus forcing users to **manually load the firmware** when initializing their wallet. Thereby, any pre-loaded keys or seeds are erased.

Effectiveness: Key attestation does prevent token replicas, if implemented with a secure CPU or secure element from which an attestation key and certificate cannot be extracted. This raises the bar for firmware modification, since attackers cannot simply flash fraudulent firmware. Forcing manual firmware loading complicates token replicas and prevents firmware modifications, given that the user overwrites fraudulent firmware with the legitimate one. This also complicates secret extraction attacks because an extracted secret would loose its value as soon as the new firmware is installed.

Usability: Manual authenticity checks are often not user-friendly. In many cases, users have to run a script via the terminal (i.e., YubiKey PIV attestation, YubiKey attestation certificates, hardware wallets' secure element authenticity check). Also, manufacturers neither sufficiently explain nor advertise these methods.

6 Survey

Our user survey was designed to address *RQ2*. In particular, we sought to answer the following questions:

- Which automatic authenticity checks are users aware of?

- Which manual authenticity checks do users perform?
- Are these authenticity checks perceived as useful?
- Do users' perceptions of security guarantees match the technical reality?

Participants who owned (i) a hardware wallet, and/or (ii) a YubiKey, and/or (iii) a smartphone were eligible to participate. They were presented with questions regarding the respective device. We recruited smartphone users as a control group to compare usage and authenticity check patterns of devices designed for security purposes only (hardware wallets, YubiKeys) with general purpose devices (smartphones). We did not include attack vectors and attestation features which solely apply to smartphones in our market review. However, all presented attack vectors (see Section 4.1) also apply to smartphones.

6.1 Discussion Rounds

Following Jensen and Laurie [47], we conducted a small-scale qualitative research study to flexibly explore the problem space before designing our survey. Two researchers did two discussion rounds with (i) a group of people working in the field of IT security who owned an HST such as a hardware wallet or a YubiKey (9 participants), and (ii) a group of people without technological expertise who owned a smartphone (3 participants). Both groups were recruited at our institution. We asked the following questions: (i) Which HSTs or devices do you own? (ii) Do you think that your hardware device was genuine when you received it? (iii) Why do you think that your hardware device was (not) genuine? (iv) Which attacks on your device can you imagine could have happened while it was distributed?

One researcher led the discussion while the other one took notes. We recorded and transcribed both discussion rounds after obtaining informed consent. Both researchers openly coded the data independently, extracting re-occurring themes and then discussing them to collect important findings for our survey design. We took the results of both discussion rounds and our market review into account when designing the main questionnaire.

6.1.1 Results (Smartphone Group)

All participants stated that they did not spend much thought on the authenticity of their device when they received it, but just assumed that it was genuine. The two most important factors influencing the participants' trust were (i) the high-quality design of the packaging, and (ii) the integrity of the stickers on the package or device. One participant stated: *"The packaging is very high quality. I'm not sure that someone who forges it [the smartphone] would put so much effort into the packaging."*

This participant further elaborated that the quality of the smartphone met expectations, i.e., the display and the buttons functioned properly. Another participant mentioned that a protection foil on the screen influenced their trust.

The participants' assessments of the likelihood of distribution attacks were mixed. One participant said: *"From the moment it [the smartphone] is in the supply chain, packaged, and this foil is on it... When you open that up, to get it all back in the same way, that is very time-consuming."*

In contrast, another participant stated: *"I can imagine that one would build something like that into the hardware, for example, spying stuff."*

6.1.2 Results (HST Group)

In contrast to the smartphone group, the majority of the HST users said that they did not fully trust the genuineness of their device when they received it. One participant explained that one can never entirely trust the cryptography on the device if one has not implemented it themself. Another participant said: *"If someone changes the hardware, there is no chance for the normal user to detect it. Especially with the Yubikey, which is cast in plastic...You can only hope you got an original key."*

Still, some participants reported that their trust in their HST was positively influenced by stickers on the packaging and by the fact that their device arrived at their home address shortly after purchasing it. One participant furthermore stated: *"I trust the Yubikey because the advertising is good and because other people I trust do trust this product."*

None of the participants opened their HST as they (i) were afraid to break it, (ii) did not want to spend time on it, or (iii) did not think that attacks based on added hardware could work. Two participants said that they checked the authenticity of their HST on the vendor's website since that was recommended in the manual. Another participant mentioned that the potential damage caused by a non-genuine device, i.e., how valuable the secrets protected by the token are, is important when deciding which authenticity checks to use. This might be a reason why the HST group invested more time and thought into the authenticity of their devices than the smartphone group.

6.2 Study Design

We opted for an online survey [56] to get a large number of—also geographically distributed—participants and, thus, quantitative insights about user perceptions and usability problems of authenticity checks deployed in HSTs. We designed our survey based on the discussion rounds and a comprehensive literature study of attack vectors. The survey consists of 25–27 closed questions (multiple-choice, 5-point Likert scale) and 2–3 open questions depending on the answers (some questions were follow-up questions). To assess the participants' security affinity, we used the Security Behavior Intentions Scale (SeBIS) [25] which quantifies intentions and self-assessments of the respondents' security behavior. We hosted the questionnaire on *Surveymonkey.com* [103]. The full questionnaire can be found on our GitHub repository [1].

If participants owned multiple eligible devices, we assigned them either to the hardware wallet sample (first choice) or the

YubiKey sample (second choice), assuming that HST users are harder to recruit than smartphone users.

6.3 Recruitment and Participants

We distributed our survey through Bitcoin, blockchain, and Yubikey mailing lists (18%), social media (75%) and personal contacts at partner institutions (7%). As compensation, we raffled gift vouchers and premium fair trade chocolates (winning chance: 6%). This approach is in line with studies by Deutskens et al. [21] and Laguilles et al. [55] which both showed that lotteries with smaller prizes but a higher winning chance are an effective strategy for increasing response rates in surveys. The demographics of our final data set are shown in Appendix 7. The sample consists mainly of male and technically adept participants, corresponding to the demographics of Bitcoin users [9] and the technology industry in general [89].

6.4 Validity and Reliability of our Dataset

To ensure sufficient statistical power, we calculated the effective sample size [61] with a significance level of .05 (95% confidence interval), and a power of .8 (the best practice value currently used [65]). These numbers yield a minimum sample size of 61 users per group. Our final dataset consists of responses from 62 hardware wallet ($\mathcal{H}$), 66 YubiKey ($\mathcal{Y}$), and 66 smartphone users ($\mathcal{S}$). We asked the participants for demographic data including their occupation and whether it is within IT security. Two thirds of our participants work in IT, from which 42% are professionally involved in IT security topics and decision-making.

We pre-tested our survey design through a think-aloud study with seven participants (non-/tech-savvy users) to check the comprehension of technical terms (taken from the manufacturers' websites) and remove biased phrasing as far as possible. Additionally, we collected expert feedback from other researchers. Our main concern was to reduce social desirability biases, especially with respect to more security-aware participants. The survey was distributed in English and German; two independent translators revised the translations. To allow unaided answers, we provided "Others" options.

In order to eliminate re-submissions and automated submission, we performed technical measures and allowed only one submission per email/IP address and device. We are confident that none of our participants lied about the possession of a hardware wallet, YubiKey, or smartphone to unfairly obtain a price in our raffle, assuming that smartphones are common. Participants who owned neither of the three devices were immediately redirected to the SeBIS [25] questions. We implemented three exclusion criteria to ensure a reliable set of data and applied them in the following order:

- *Four open and two check-up questions* (re-phrasing earlier questions or providing invalid answer possibilities), which we manually checked for consistency and meaningfulness (21 participants were removed).
- *One attention check question* with shuffled answer options (58 participants were removed).
- *Completing of the questionnaire* was mandatory (six participants were removed).

In total, 279 participants took part in our survey. After applying our exclusion criteria, we reached a final sample of $n = 194$ for our analysis.

6.5 Data Analysis

Besides descriptive statistics, we also performed statistical tests. For closed-ended nominal scaled questions, we conducted pair-wise χ^2 tests between our three groups and interpreted the effect size Cramér's V [49]. In cases where the expected frequencies were smaller than 5, we additionally conducted a Fishers' Exact test. To counteract the multiple comparisons problem for multiple answer questions, we applied the Holm–Bonferroni correction [41]. For interval-scaled questions, we calculated the Pearson correlation coefficient ρ. We rejected the null-hypothesis of independence when p was smaller than .05 (95% confidence interval).

Regarding the open questions (qualitative data), two researchers independently coded the responses concerning (i) the improvement suggestions of authenticity checks, and (ii) the "other" answer option to closed-ended questions. We created a codebook, coded the entire data and discussed conflicts until agreement was reached among the coders. Our inter-rater reliability $\alpha = .91$ (Krippendorff's Alpha value [54]) indicates a high level of agreement.

6.6 Ethical Considerations

Our ethical review board approved the study. Preserving the participants' privacy and limiting the collection of sensitive information as far as possible are fundamental principles. We assigned the study participants IDs to anonymously process their data. The collected email addresses from raffle participants were stored separately from the survey responses. All participants were informed about the data handling procedures and gave informed consent. The study strictly followed the EU's General Data Protection Regulation (GDPR).

6.7 Results

6.7.1 Device Usage (Q2, Q3, Q18)

We observed significant differences in the device usage across all groups ($\chi^2(\mathcal{YH}, \mathcal{YS}, \mathcal{HS}) : p < .02$) with high V for $\mathcal{HS}$ (.6) and $\mathcal{YS}$ (.42) and a medium V for $\mathcal{YH}$ (.28). Only 45% of $\mathcal{H}$ and 66% of $\mathcal{Y}$, but 98% of $\mathcal{S}$ use their devices regularly. We attribute this to the fact that smartphones are commonly used for everyday tasks and HSTs for security-related tasks only. Moreover, we explain the more frequent device usage of $\mathcal{Y}$ as opposed to $\mathcal{H}$ by the fact that authentication tasks are performed more often than cryptocurrency transactions.

Related to the usage context, we found significant differences with large V between $\mathcal{H}$ and the other two groups

($\chi^2(\mathcal{YH},\mathcal{HS})$: $p < .01$, $V > .39$ [large]). In contrast, no notable differences emerged between $\mathcal{Y}$ and $\mathcal{S}$. The majority of $\mathcal{Y}$ and $\mathcal{S}$ reported to use their devices in both their private and professional life ($\mathcal{Y}$:50%, $\mathcal{S}$:47%), followed by exclusively private ($\mathcal{Y}$:39%, $\mathcal{S}$:48%) or professional usage ($\mathcal{Y}$:10%, $\mathcal{S}$:5%). In comparison, 85% of $\mathcal{H}$ stated to use their HST only for private purposes with much lower percentages for private and professional (11%) or only professional usage (3%).

6.7.2 Trust Factors (Q4/a, Q17)

The upper part of Table 3 shows whether our participants trusted their devices' genuineness when receiving it. Most of $\mathcal{H}$ and $\mathcal{S}$ stated that they did, whereas $\mathcal{Y}$ were more sceptical and often reported a lack of knowledge ($\chi^2(\mathcal{YH})$: $p < .03$, $V > .19$ [small]; $\chi^2(\mathcal{YS})$: $p < .03$, $V > .22$ [medium]). We assume this difference might be due to recent media reports on flawed or counterfeit YubiKeys [16, 97].

The lower part of Table 3 describes which factors influenced the participants' trust in the genuineness of their devices when they received it. Generally, the majority of all groups reported a high influence of the packaging characteristics on their trust, except for holographic stickers which mostly did not affect their perception of device genuineness. There were significant differences in trust in the vendors name and logo between $\mathcal{H}$ and $\mathcal{S}$ ($\chi^2(\mathcal{HS}): p < .01$, $V = .29$ [small]) and noticeable although insignificant differences between $\mathcal{Y}$ and $\mathcal{S}$. Moreover, significant differences emerged in regards to high-quality packaging between $\mathcal{Y}$ and $\mathcal{S}$ ($\chi^2(\mathcal{YS}): p < .01$, $V = .3$ [medium]). This shows that HST users are more sceptical about these packaging characteristics than $\mathcal{S}$.

An undamaged product increased the trust of the majority of all groups with no significant differences across them. In contrast, less than half of the participants found it important that their device was not put into operation with significant differences between $\mathcal{Y}$ and $\mathcal{H}$ ($\chi^2(\mathcal{YH}): p < .01$, $V = .25$ [small]). The higher numbers for $\mathcal{H}$ can be explained with reports on attacks utilizing pre-initialized hardware wallets [98]. The majority of participants trust the manufacturer. Although not significant, the opinion of other people is more important for HST users than for $\mathcal{S}$.

6.7.3 Performed Authenticity Checks (Q5)

After receiving their devices, $\mathcal{H}$ performed the most and $\mathcal{S}$ the least authenticity checks (see Figure 2). We attribute these low $\mathcal{S}$ numbers to a lacking "authenticity check culture" in the smartphone world, as smartphones are not solely designed for security purposes. Most smartphone manufacturers do not offer any form of authenticity checks. Furthermore, we explain the fact that $\mathcal{H}$ are more willing to perform authenticity checks than $\mathcal{Y}$ with potentially highly valuable monetary assets stored on $\mathcal{H}$ tokens.

More $\mathcal{H}$ than $\mathcal{S}$ compared the outside of their devices with reference pictures ($\chi^2(\mathcal{HS}): p < .01, V = .33$ [medium]). Moreover, HST users performed checks on the manufacturers websites more often than $\mathcal{S}$ ($\chi^2(\mathcal{HS}): p < .01, V = .54$ [large], $\chi^2(\mathcal{YS}): p < .01, V = .42$ [medium]). $\mathcal{H}$ conducted the most software signature checks with significant differences to $\mathcal{S}$ ($\chi^2(\mathcal{HS}): p < .01, V = .36$ [medium]).

Table 3: Trust Factors of Token Genuineness (Selection)

			$\mathcal{H}$	$\mathcal{Y}$	$\mathcal{S}$
Genuine		Yes	95%	82%	93%
		No	0%	0%	3%
		I don't know	5%	17%	5%
Trust Factors	Packaging	not damaged/opened	74%	65%	77%
		vendors name/logo displayed	45%	56%	75%
		high quality	47%	33%	64%
		holographic sticker	33%	31%	34%
	Product	not damaged	65%	70%	79%
		has not been put into operation	40%	12%	36%
		looked genuine	66%	73%	83%
	Trusted	manufacturer	73%	61%	61%
		other people's opinion	63%	68%	50%

Groups: Hardware Wallet ($\mathcal{H}$), YubiKey ($\mathcal{Y}$), and Smartphone users ($\mathcal{S}$). For the trust factors, multiple answers were possible.

A significant higher percentage of $\mathcal{Y}$ than $\mathcal{H}$ reported to not have performed any authenticity check at all ($\chi^2(\mathcal{HY}): p < .01, V = .30$ [medium]). Since YubiKeys cannot be opened, $\mathcal{Y}$ did not inspect their device's interior (check-up question: $\chi^2(\mathcal{HY}): p < .01, V = .28$ [small], $\chi^2(\mathcal{HS}): p < .01, V = .28$ [small]). We did not specifically ask $\mathcal{H}$ about their hardware wallet model, but about the manufacturer and the year of purchase. Based on this information we estimate that more than 50% of $\mathcal{H}$'s hardware wallets can be opened. However, less than half of those performed this check. Two participants explicitly stated they were afraid to damage the case.

6.7.4 Manual and Automatic Checks (Q8, 10, 15, 16)

About half of HST owners ($\mathcal{Y}$:41%, $\mathcal{H}$:51%) and 3% of $\mathcal{S}$ reported that they performed manual checks, which is in line with their answers to Q5 (see Section 6.7.3). We asked our participants whether authenticity check instructions were provided by the manufacturers and found significant differences in their answers ($\chi^2(\mathcal{YH},\mathcal{YS},\mathcal{HS})$: $p < .01$, $V > .35$ [large]). 72% of $\mathcal{H}$, but only 30% of $\mathcal{Y}$ and 6% of $\mathcal{S}$ stated that instructions were provided. From those participants who

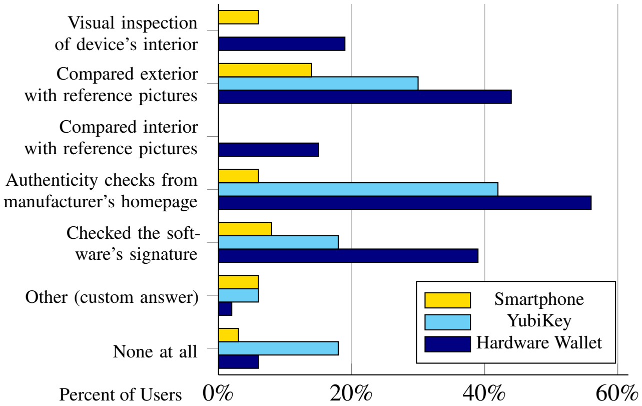

Figure 2: Performed Authenticity Checks (Self-Reported)

Table 4: Perceived Security of Authenticity Checks (Selection)

		not swapped	no SW altered	no HW manipulated	no HW added	none	I don't know
Holographic sticker	$\mathcal{H}$	31%	23%	26%	19%	**39%**	3%
	$\mathcal{Y}$	39%	9%	26%	17%	**45%**	6%
	$\mathcal{S}$	**36%**	17%	24%	17%	24%	20%
Interior inspection	$\mathcal{H}$	**34%**	5%	19%	**34%**	24%	32%
	$\mathcal{Y}$	14%	3%	12%	9%	29%	**43%**
	$\mathcal{S}$	**27%**	5%	24%	**27%**	18%	**26%**
Automatic checks	$\mathcal{H}$	49%	**74%**	38%	22%	2%	0%
	$\mathcal{Y}$	54%	**89%**	11%	0%	0%	0%
	$\mathcal{S}$	**31%**	17%	9%	17%	6%	5%
Manual checks	$\mathcal{H}$	32%	**56%**	41%	22%	2%	0%
	$\mathcal{Y}$	33%	**48%**	33%	30%	3%	0%
	$\mathcal{S}$	–	–	–	–	–	–
Signature or hash check	$\mathcal{H}$	24%	**82%**	17%	13%	2%	8%
	$\mathcal{Y}$	21%	**75%**	15%	12%	3%	5%
	$\mathcal{S}$	27%	**65%**	17%	11%	6%	17%
Single-piece cast	$\mathcal{H}$	8%	5%	**29%**	**31%**	**32%**	23%
	$\mathcal{Y}$	9%	6%	**41%**	**41%**	33%	11%
	$\mathcal{S}$	–	–	–	–	–	–

Hardware wallet ($\mathcal{H}$), YubiKey ($\mathcal{Y}$), and Smartphone users ($\mathcal{S}$).
– Manual checks and single piece casts are mostly not applicable for $\mathcal{S}$

answered to Q15 with "yes" (prerequisite for answering Q16), the majority ($\mathcal{H}$:80%, $\mathcal{Y}$:70%) thinks that they have carried out all provided checks. Thereby, we observed no significant differences between $\mathcal{Y}$ and $\mathcal{H}$ ($\chi^2(\mathcal{YH}) : p > .08, V = .35$ [large]). The answers of $\mathcal{S}$ to Q15 can be neglected, since only four participants of $\mathcal{S}$ answered that their manufacturer provided information on manual authenticity checks and, hence, reached this question.

Our results show significant differences between all groups in their assessment of whether automatic authenticity checks are performed by their devices ($\chi^2(\mathcal{YH},\mathcal{YS},\mathcal{HS}) : p < .02, V > 0.24$ [medium]). 76% of $\mathcal{H}$ were confident that automatic checks are implemented in contrast to 14% of $\mathcal{Y}$ and 21% of $\mathcal{S}$. Some stated that they do not know whether any checks were carried out ($\mathcal{H}$:13%, $\mathcal{Y}$:29%, $\mathcal{S}$:45%), which indicates a lack of information material as the automatism might conceal this method's existence.

6.7.5 Perceived Security (Q6, Q7, Q9, Q11, Q12, Q13)

We asked our participants about their perceived effectiveness of authenticity checks in relation to attack vectors (Table 4).

Holographic Stickers: Their ineffectiveness, as found in the market review, was perceived as such by about two-fifths of HST users. Half of them mistakenly stated that these stickers prevent token replication or hardware/software modifications. In contrast, $\mathcal{S}$ mainly reported prevention against swapping devices, and only a fourth of them attributed no effectiveness to the stickers. A higher percentage of $\mathcal{S}$ than HST users reported a lack of knowledge with significant differences between $\mathcal{S}$ and $\mathcal{H}$ ($\chi^2(\mathcal{HS}) : p < .01, V = .25$ [small]).

Interior Inspection: The majority of $\mathcal{Y}$ reported a lack of knowledge on its effects, which we attribute to the single-piece cast of YubiKeys that prevent interior inspection. This led to significant differences between $\mathcal{Y}$ and $\mathcal{H}$ for added hardware ($\chi^2(\mathcal{YH}) : p < .04, V = .30$ [medium]) and noticeable although not significant differences for swapped devices. In fact, both attacks are made more difficult through interior inspection. A large fraction of $\mathcal{H}$ reported a lack of knowledge or ineffectiveness of the method, which suggests missing information material. Two $\mathcal{H}$ participants reported that they refrained from opening their token out of fear to damage it.

Automatic Checks: Large fractions of $\mathcal{H}$ and $\mathcal{Y}$ indicated that automatic checks prevent software modification, which is correct (see Table 1). About one-third of $\mathcal{S}$ and about half of $\mathcal{Y}$ and $\mathcal{H}$ thought that automatic checks prevent token replication. This is partly true if remote firmware attestation is implemented, which can complicate token replication. Moreover, many $\mathcal{H}$ and some $\mathcal{Y}$ stated that automatic checks would prevent hardware modifications or added hardware, which is incorrect. These results show that the benefits of automatic checks are not clearly communicated to (HST) users.

Manual Checks: $\mathcal{Y}$ and $\mathcal{H}$ stated that manual checks mainly prevent software modification, followed by hardware manipulation, token replication, and additional hardware. For $\mathcal{S}$, not enough participants reported having performed these checks in order to draw statistically significant conclusions. Depending on the manual checks offered, these can indeed prevent software modification (manual firmware load, firmware attestation) or token replication (key attestation). However, manual authenticity checks cannot protect against added hardware or IC modifications. From these results, we derive that HST users do not have a clear idea of the security benefits offered by manual checks. This is also emphasized by the results of Q16a where HST users stated that they did not know which authenticity checks existed and were not sure whether these methods are suitable to ensure genuineness. Although manual and automatic checks can offer equal protection, our participants trusted the automatic checks more. This indicates that especially the benefits of manual checks are not sufficiently conveyed to them.

Signature or Hash Check: The majority of all groups correctly reported that this check prevents software modification, with no significant differences across groups. Nevertheless, many participants (additionally) incorrectly stated that it would protect against swapped devices, hardware implants or modification. Participants of all groups assessed the benefits of automatic/manual software checks differently in comparison to signature/hash verification—although they are actually similar—and assumed a connection between software checks and hardware authenticity.

Single-Piece Cast: Although some participants correctly stated that single-piece casts can prevent (or complicate) hardware implants/modification (see Table 1), a third reported that

Table 5: Perceived vs. Actual Effectiveness of Attestation

Actual effectiveness (market review)
○ no prevention ● strong protection
◐ complicates attack/decreases usefulness

Perceived effectiveness (survey)
0% 100%

over-estimated under-estimated[1]

Attestation / Countermeasure			Hardware implants	Token replication	IC modification	Firmware modification	USB exploit	Token pre-initialization	Timing side-channels	Bus snooping	IC microprobing	Fault injection
			Hardware			Software			Secret Extraction			
Attestation / Countermeasure	Pack.	Tamper-evident	◐	◐	◐	◐	◐	◐	◐	◐	◐	◐
		Holographic sticker	○	○	○	○	○	○	○	○	○	○
	Hardware	Single-piece cast	●	◐	◐	○	○	○	○	◐	◐	○
		Openable device	◐	◐	○	○	○	○	○	○	○	○
		Secure element (co-processor)	○	●	●	○	○	○	●	◐	●	●
		Secure CPU	◐	●	●	○	○	○	●	●	●	●
	Software	Local firmware attestation	○	○	○	●	○	○	○	○	○	○
		Remote firmware attestation	○	◐	○	●	○	○	○	○	○	○
		Key attestation	○	●	○	◐	○	○	○	○	○	○
		Manual firmware load	○	◐	○	●	○	◐	◐	◐	◐	◐

[1] Benefits need to be better explained to customers.

they do not have any benefit. Generally, $\mathcal{Y}$ rated the benefits higher than $\mathcal{H}$ with significant differences for adding hardware $\chi^2(\mathcal{Y}\mathcal{H}) : p < .01, V = .42$ [medium]). We assume this is the case because hardware wallets are not cast in one piece. However, our findings show that also $\mathcal{Y}$ underestimate the security benefits offered by single-piece casts.

6.7.6 Perceived Likelihood of Attacks (Q14)

In order to determine whether our participants feel the need for authenticity checks, we asked how they perceive the likelihood of different attacks (see Appendix 3). All groups consider the presented attack vectors as unlikely. For the extraction of valuable information, we observed slightly higher percentages of "very likely" answers, which we attribute to frequent media reports on such attacks [24, 74].

6.7.7 Security Awareness (Q23)

Based on nine questions from the Security Behavior Intentions Scale (SeBIS) by Egelman and Peer [25], we conclude that our sample is more security-conscious (score: $\mu > 4.1$, $\sigma = 1.1$) than their sample. The SeBIS answers of $\mathcal{Y}$ and $\mathcal{H}$ correlate strongly with each other ($\rho_{\mathcal{H}\mathcal{Y}} > .97$, $p < .02$) and less with $\mathcal{S}$ ($\rho_{\mathcal{H}\mathcal{S}} \approx .91$, $\rho_{\mathcal{Y}\mathcal{S}} \approx .9$, $p < .04$) as our $\mathcal{S}$ group is slightly less security-savy.

We found no statistical significance for a correlation between the participants SeBIS score and correct or incorrect assessments of the authenticity checks. This might be due to the fact that our participants' SeBIS answers are very homogeneous, and most of them are in the upper third of the SeBIS.

6.8 Limitations

Our recruitment methods allowed us to collect a representative sample from the IT industry and Bitcoin community (see Section 6.3), which is skewed towards male and tech-savvy participants. Due to the lack of available data regarding the overall population of hardware wallet and YubiKey users we cannot conclude whether our sample is representative of these two groups with respect to demographics. Moreover, no expressive evaluation of cultural differences can be made as our sample is biased towards residents of industrial countries. As the survey was conducted either in English (87%) or German (13%), people speaking other native languages and/or having no or weak English or German skills were potentially excluded. Furthermore, if longer periods of time have passed between HST purchase and taking part in the study, this may have influenced how accurately the participants recalled the authenticity checks.

7 Discussion

In this section we connect the results from the survey and the market review and provide seven actionable recommendations for a secure and user-friendly design of HST authenticity checks (answering *RQ3*). Table 5 compares the effectiveness of attestation methods as assessed in the market review with user perceptions. Darker shades of red indicate methods that users perceive as effective, while lighter shades indicate perceived ineffectiveness. These colors visualize trends based on the HST users' responses to whether the described method prevents certain attacks (Q4a, Q6, Q7, Q9, Q11–Q13). The effectiveness of some methods is correctly perceived, while that of others is severely over-estimated, severely under-estimated, or slightly over-estimated (see the following subsections). Our *survey* showed that these gaps between perceived and actual security of HST authenticity checks are caused by a lack of information and transparency. Our *market review* also revealed that many manufacturers do not implement sufficient and/or transparent authenticity checks. We found that no single check and no currently implemented combination of checks can adequately protect against the multitude of attack vectors described in this paper. In order to maximize the security and usability of HST attestation, we suggest (i) a **user-centered transparent design** combined with (ii) **secure CPUs or elements**, (iii) **remote firmware attestation**, and (iv) **collaborative protocols** [19].

7.1 User-centered Design

To close the gap between the perceived and actual security of authenticity checks, we propose to visualize the availability and results of authenticity checks (incl. information material) and provide standardized labels.

7.1.1 Transparent authenticity checks

In line with Distler et al. [22], Fahl et al. [29], Mathiasen et al. [69], and Mai et al. [67], we argue that security mechanisms should not be completely hidden from users since transparency of these mechanisms is crucial for perceived and actual security. In order to understand security characteristics and build up trust, authenticity checks should be visible to users by adapting the UI of HSTs. Rode et al. [90] showed that visualizations of system aspects can help people understand a system's security and incorporate it into their actions. They suggest not to overwhelm users with complexity but dynamically visualize system aspects as a GUI's temporal response. For HSTs, this could be achieved by spinners, check-marks, or warnings that explain the process and results of authenticity checks. The effectiveness of security warnings was demonstrated by Akhawe et al. [4] who also recommended that security information should be communicated to users.

Our market review found a lack of accessible information on implemented authenticity checks, which was confirmed by our survey. Our survey also indicates that users can be effectively reached via online or offline information.

Recommendation 1: *The existence and results of authenticity checks should be visualized to users (e.g., through spinners, check-marks, or warnings) following the respective design literature [100, 102, 112]. Easily accessible descriptions of authenticity checks should be provided online and offline.*

7.1.2 Security Labels

To avoid overwhelming users with complex information, security labels similar to the European Union energy label [15], with equivalent grades from *A* (most secure) to *G* (least secure), should be introduced. Clear guidelines for manufacturers on how to achieve a specific grade should be provided. Such labels could encourage manufacturers to implement high-quality and secure authenticity checks by giving them a competitive advantage. Our evaluation framework (attack vectors) and recommendations provide a basis for constructing such labels and to certify the implementation of secure HST design rules.

Available independent certifications, such as NIST FIPS 140-2 [77], were explicitly designed for cryptographic modules in governmental use. They are not easily understandable for end users and do not cover attestation mechanisms. To obtain such a certificate, third-party laboratories have to conduct costly tests. Such certificates should be taken into account when creating user-friendly HST security labels.

Recommendation 2: *Self-explanatory security labels should be placed on the HSTs to facilitate purchase decisions.*

7.2 Secure CPU or Secure Elements

Protecting the key material in trustworthy hardware is paramount for HSTs. The effectiveness of many software methods (including local/remote software attestation and key attestation) is dramatically increased when implemented on a side-channel-hardened circuitry (see Section 5.2.3). Nevertheless, only six out of the nine tested devices use secure CPUs or secure elements. We think this is due to additional costs and effort during design and production. In general, secure CPUs and secure elements do not require user interaction. However, in our survey we found that their effectiveness is incorrectly assessed by the majority of the users as their existence and benefits are not transparent. Hence, secure elements/CPUs cannot affect users' trust.

Recommendation 3: *HSTs should deploy a secure element or secure CPU that contains critical operations and data and checks the firmware in a secure boot setup. Authenticity checks should be visible, and security labels should verify their existence (see Section 7.1).*

7.3 Remote Firmware Attestation

Remote attestation (currently only implemented in two of the tested devices) is more effective than local methods. Our survey showed that the effectiveness of local and remote firmware attestation is assessed correctly (see Table 5). However, we also found that many users are not aware of these methods. Hence, they do not increase the users' trust in their devices.

Recommendation 4: *Methods for remote firmware attestation should be implemented. These methods should be made visible to users (see Section 7.1).*

7.4 Collaborative Protocols

In-protocol secret leakage and weak or pre-loaded keys on HSTs can be prevented with collaborative protocols. To date, no off-the-shelf HST implements this approach. Thereby, the single point of computation at the HST is removed and distributed equally between the HST and the browser. The key and signatures are generated during their interaction; thereby, the browser can enforce the HST's correct behavior without learning the secret. If one of the parties produces attacker-impacted results, the other one will detect that.

Dauterman et al. [19] recently showed that collaborative key and signature generation is feasible for U2F tokens. It is not straightforward to apply such protocols to hardware wallets, as confirmed by one manufacturer. Hardware wallets commonly use BIP32 (a standard for hierarchical deterministic wallets) [58, 111] which does not support collaborative and verifiable key generation building on verifiable identity families (VIFs). Upgrading hardware wallets to use such protocols would force users to distinguish between BIP32 and upgraded wallets, given that the key schemes are different.

Collaborative protocols have no immediate usability issues since the execution is hidden from the users. However, secrets can still be leaked out-of-band or via the USB interface, and ransom attacks are still feasible.

Recommendation 5: *U2F tokens should implement collaborative protocols for key and signature generation. Other token families should consider a long-term switch. Self-explanatory labels and sufficient information material (see Section 7.1) should be used to inform users about implemented methods and achieved security benefits.*

7.5 Manual vs. Automated Checks

Our survey shows that many HST users are not aware of the performed automatic checks or underestimate their effectiveness (key attestation) due to a lack of visibility and information. Our market review and survey revealed that manual authenticity checks are often too complicated or time-consuming for users and, moreover, are not sufficiently advertised by manufacturers (e.g., command line checks). Hence, they are only performed by half of $\mathcal{H}$ and less than half of $\mathcal{Y}$. We found that many users are eager to compare the packaging or the product with reference pictures, but felt let down by manufacturers who only provided insufficient material.

Recommendation 6: *HSTs should implement automated but transparent authenticity checks (see Section 7.1). If manual methods are used, they should be a mandatory part of the initialization process. Moreover, all methods should be easily visible and explained on the manufacturers' websites.*

7.6 Openable Devices vs. Single-Piece Casts

Token manufacturers use two mutually exclusive approaches: (i) easily openable HSTs for visual inspection of the interior, or (ii) unopenable HSTs with integrated electronics or a single-piece cast. Both approaches are double-edged swords:

Our market review shows that easily *openable devices* provide quick access to users and attackers alike. However, our survey also revealed that users rarely open and visually inspect their HST. A possible remedy would be an application that automatically compares pictures of the devices' inside to reference pictures. Still, verifying the genuineness of such an application would pose a new and complex challenge.

On the other hand, users can easily inspect the state of *single-piece cast* devices, which provides some level of security assertion. The inside cannot be seen; therefore, an attacker needs to create a similar-looking outer appearance of the product (see Section 5.2.2). Our survey showed that users incorrectly rate the security benefits of single-piece cast devices rather low. Manufacturers could experiment with see-through molded devices; the Ledger Nano S series already offers transparent cases. However, it is also openable, thus diminishing the security intent of the transparent case.

Recommendation 7*: If manual inspection of the hardware is required, it should be tightly integrated in the initialization process. If single-piece casts are used, their security properties should be clearly communicated to the user (Section 7.1).*

7.7 Security Theater

Security theater [96] describes actions aiming at creating a sense of security although they are not (or only marginally) effective. In our market review, we could verify that although holographic stickers are frequently used, they are ineffective against all attack vectors presented in this work. Our survey confirms that, depending on the target audience, many customers understood the insignificance of holographic stickers, while others assumed that they offer a level of protection. Along these lines, e.g., Ledger claims not to use anti-tampering seals (or holographic seals) since they give users a false sense of security [57].

We also found that it is common practice to utilize tamper-evident packaging, although this is less effective than other approaches. Our survey suggests that packaging profoundly influences the users' trust. We recognize that this method might be useful to increase people's trust in their HST. However, we argue that such an approach on its own is unrewarding in the long-term.

Recommendation 8: *Authenticity checks that give users a false feeling of security while only being marginally (or not) effective should be disestablished.*

8 Conclusion

Our findings show that technical and usability issues of authenticity checks in widely used HSTs undermine the security benefits these tokens are supposed to provide. We performed a market review of state-of-the-art HSTs and a large-scale survey to assess users' perceptions and usage of authenticity checks. Our results suggest that commonly used authenticity checks—even the best-case implementations—are not sufficient to defeat distribution attacks. Moreover, users cannot make informed trust decisions based on the deployed methods as their existence and benefits are often hidden. Thus, users currently base their trust decisions to a large extent on visual but noneffective features such as packaging.

Based on our findings, we suggest a multi-faceted approach maximizing security through automation and user engagement. We propose more transparency for users, secure elements/CPUs, and collaborative protocols for practical improvement.

As future work, further usability studies will be required to determine these suggestions' actual effectiveness and efficiency. For instance, lab studies could be conducted in which counterfeit/modified HSTs, with and without our recommended improvements, are provided to users in order to assess how well they can detect attacks, and how much of a difference can be achieved through our suggestions. We furthermore aim at developing a standardized procedure, including security labels which help users to judge HST genuineness based on the implemented authenticity checks. We will use participatory design studies to investigate how security guarantees can be visualized and made transparent to users. We also plan to examine novel attestation methods (e.g., transparent molded enclosures) via user experiments. Researchers should also study how collaborative protocols and currently not implemented hardware tampering prevention approaches (e.g., sensors, hardware metering) can be applied to HSTs.

We propose to include prospective users in the design process of HSTs to determine a threshold for user engagement and obfuscation. Our work strongly suggests that the research community, together with token manufacturers, needs to assess and develop secure yet comprehensible authenticity checks for HSTs.

Acknowledgements

We thank the anonymous reviewers, our shepherd Blase Ur as well as Michael Schwarz, Christian Kudera, Manuel Wiesinger and Sven Bugiel for their valuable feedback on our work.

This material is based upon work partially supported by the Austrian Research Promotion Agency's (FFG) *Industry-related Dissertation* funding programm and the United States Office of Naval Research (ONR) under contract N00014-17-1-2782. We also gratefully acknowledge an ENDEAVOR award from the Donald Bren School of Information and Computer Sciences at UC Irvine. SBA Research (SBA-K1) is a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the province of Vienna. The COMET Programme is managed by FFG.

References

[1] Auxiliary material to "On the usability of authenticity checks for hardware security tokens" paper. https://github.com/adriandab/usec-hwtoken.

[2] C. Z. Acemyan, P. Kortum, J. Xiong, and D. S. Wallach. 2FA Might Be Secure, But It's Not Usable: A Summative Usability Assessment of Google's Two-factor Authentication (2FA) Methods. In *The Human Factors and Ergonomics Society Annual Meeting*, 2018.

[3] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using IC fingerprinting. In *IEEE Symposium on Security and Privacy*, 2007.

[4] D. Akhawe and A. P. Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *USENIX Security Symposium*, 2013.

[5] R. Anderson. *Security engineering: A Guide to Building Dependable Distributed Systems*. 2008. ISBN 978-0470068526.

[6] F. Armknecht, A.-R. Sadeghi, S. Schulz, and C. Wachsmann. A Security Framework for the Analysis and Design of Software Attestation. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1–12, 2013.

[7] A. Baumgarten, M. Steffen, M. Clausman, and J. Zambreno. A case study in hardware trojan design and implementation. *International Journal of Information Security*, 10(1), 2011.

[8] S. Bhunia and M. Tehranipoor. *Hardware Security: A Hands-on Learning Approach*. Morgan Kaufmann, 2018. ISBN 978-0128124772.

[9] J. Bohr and M. Bashir. Who uses Bitcoin? An exploration of the Bitcoin community. In *IEEE International Conference on Privacy, Security and Trust (PST)*, 2014.

[10] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE Symposium on Security and Privacy*, 2012.

[11] M. Brengel and C. Rossow. Identifying key leakage of Bitcoin users. In *International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2018.

[12] L. Champine. A Ransom Attack on Hardware Wallets, 2019. https://blog.sia.tech/534c075b3a92, accessed: 2020-06-17.

[13] Y.-S. Choi, Y.-J. Jeon, and S.-H. Park. A study on sensor nodes attestation protocol in a wireless sensor network. In *International Conference on Advanced Communication Technology*, 2010.

[14] S. Ciolino, S. Parkin, and P. Dunphy. Of two minds about two-factor: Understanding everyday FIDO U2F usability through device comparison and experience sampling. In *Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[15] E. Commission. About the energy label and ecodesign. https://ec.europa.eu/info/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/energy-label-and-ecodesign/about_en, accessed: 2020-06-07.

[16] J. Cox. Hackers Show Proofs of Concept to Beat Hardware-Based 2FA, 2014. https://www.vice.com/en_us/article/8xazek/hackers-show-proof-of-concepts-to-beat-hardware-based-2FA, accessed: 2020-06-10.

[17] A. Dabrowski, H. Hobel, J. Ullrich, K. Krombholz, and E. Weippl. Towards a hardware trojan detection cycle. In *International Workshop on Emerging Cyberthreats and Countermeasures, ECTCM*, 2014.

[18] S. Das, A. Dingman, and L. J. Camp. Why Johnny Doesn't Use Two Factor A Two-Phase Usability Study of the FIDO U2F Security Key. In *International Conference on Financial Cryptography and Data Security*, 2018.

[19] E. Dauterman, H. Corrigan-Gibbs, D. Mazières, D. Boneh, and D. Rizzo. True2F: Backdoor-resistant authentication tokens. In *IEEE Symposium on Security and Privacy*, 2019.

[20] S. Dechand, D. Schürmann, K. Busse, Y. Acar, S. Fahl, and M. Smith. An empirical study of textual key-fingerprint representations. In *USENIX Security Symposium*, 2016.

[21] E. Deutskens, K. De Ruyter, M. Wetzels, and P. Oosterveld. Response rate and response quality of internet-based surveys: An experimental study. *Marketing letters*, 15(1), 2004.

[22] V. Distler, M.-L. Zollinger, C. Lallemand, P. B. Roenne, P. Y. A. Ryan, and V. Koenig. Security - Visible, Yet Unseen? In *ACM Conference on Human Factors in Computing Systems*, 2019.

[23] Q. Dong and G. Xiao. A subliminal-free variant of ECDSA using interactive protocol. In *International Conference on E-Product E-Service and E-Entertainment*, 2010.

[24] A. Drozhzhin. How to hack a hardware cryptocurrency wallet. https://www.kaspersky.co.uk/blog/hardware-wallets-hacked/15154/, accessed: 2020-06-17.

[25] S. Egelman and E. Peer. Scaling the Security Wall: Developing a Security Behavior Intentions Scale (SeBIS). In *ACM Conference on Human Factors in Computing Systems*, 2015.

[26] J. Ehrensvärd, Y. J. Kemp, and F. Alliance. FIDO U2F HID protocol specification, 2014. https://fidoalliance.org/specs/fido-u2f-v1.0-ps-20141009/fido-u2f-hid-protocol-ps-20141009.html.

[27] K. Eldefrawy, G. Tsudik, A. Francillon, and D. Perito. SMART: Secure and Minimal Architecture for (Establishing Dynamic) Root of Trust. In *The Network and Distributed System Security Symposium*, 2012.

[28] S. Eskandari, J. Clark, D. Barrera, and E. Stobert. A first look at the usability of Bitcoin key management. In *NDSS Symposium USEC Workshop*, 2015.

[29] S. Fahl, M. Harbach, T. Muders, M. Smith, and U. Sander. Helping Johnny 2.0 to encrypt his Facebook conversations. In *Symposium on Usable Privacy and Security*, 2012.

[30] Feitian. U2F Keys, 2020. https://www.ftsafe.com/Products/FIDO/Single_Button_FIDO, accessed: 2020-09-16.

[31] D. Forte, C. Bao, and A. Srivastava. Temperature tracking: An innovative run-time approach for hardware trojan detection. In *IEEE/ACM International Conference on Computer-aided Design*, 2013.

[32] E. Gelenbe and Y. M. Kadioglu. Energy life-time of wireless nodes with network attacks and mitigation. In *IEEE International Conference on Communications Workshops*, 2018.

[33] A. Gkaniatsou, M. Arapinis, and A. Kiayias. Low-Level Attacks in Bitcoin Wallets. In P. Q. Nguyen and J. Zhou, editors, *Information Security*, 2017.

[34] Google. Google Trends - U2F Token Analysis, 2019. https://trends.google.com/trends/explore?q=yubikey,u2f%20token,thetis,hyperfido, accessed: 2020-06-17.

[35] Google. Titan Security Key, 2020. https://store.google.com/product/titan_security_key, accessed: 2020-09-16.

[36] J. Grand and G. I. Studio. Understanding hardware security. *Black Hat Japan*, 2004. https://www.blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-grand/bh-JP-04-grand.pdf.

[37] Y. Grauer. The best security key for multi-factor authentication, 2020. https://www.nytimes.com/wirecutter/reviews/best-security-keys/, accessed: 2020-09-15.

[38] U. Guin, P. Cui, and A. Skjellum. Ensuring proof-of-authenticity of IoT edge devices using blockchain technology. In *IEEE International Conference on Blockchain*, 2018.

[39] K. He, X. Huang, and S. X.-D. Tan. Em-based on-chip aging sensor for detection of recycled ics. *IEEE Design & Test*, 33(5):56–64, 2016.

[40] HexView. Inside Yubikey Neo, 2018. http://www.hexview.com/~scl/neo/, accessed: 2020-06-17.

[41] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 1979.

[42] H.-C. Hsiao, Y.-H. Lin, A. Studer, C. Studer, K.-H. Wang, H. Kikuchi, A. Perrig, H.-M. Sun, and B.-Y. Yang. A Study of User-Friendly Hash Comparison Schemes. In *Annual Computer Security Applications Conference*, Dec 2009.

[43] A. Huang. Supply Chain Security: "If I were a Nation State...", 2019. BlueHat IL, video available: https://youtu.be/RqQhWitJ1As, accessed: 2020-06-17.

[44] T. Idriss and M. Bayoumi. Lightweight highly secure puf protocol for mutual authentication and secret message exchange. In *IEEE International Conference on RFID Technology & Application (RFID-TA)*, 2017.

[45] Intertronix. Custom Hologram Sticker Online, 2020. https://www.intertronix.com/category-s/1673.htm, accessed: 2020-09-16.

[46] J. Jang, J. Woo, J. Yun, and H. K. Kim. Mal-netminer: malware classification based on social network analysis of call graph. In *International Conference on World Wide Web*, 2014.

[47] E. Jensen and C. Laurie. *Doing real research: A practical guide to social research*. 2016. ISBN 978-1446273883.

[48] R. G. Johnston. Tamper-indicating seals. *American Scientist*, 94(6), 2006.

[49] H.-Y. Kim. Statistical notes for clinical researchers: chi-squared test and Fisher's exact test. *Restorative dentistry & endodontics*, 42(2), 2017.

[50] A. Kingsley-Hughes. Best security keys in 2020: Hardware-based two-factor authentication for online protection. https://www.zdnet.com/article/best-security-keys/, accessed: 2020-09-15.

[51] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, et al. Spectre attacks: Exploiting speculative execution. In *IEEE Symposium on Security and Privacy*, 2019.

[52] F. Koushanfar. Hardware metering: A survey. In

Introduction to Hardware Security and Trust. 2012. ISBN 978-1-4419-8080-9.

[53] C. Krieg, A. Dabrowski, H. Hobel, K. Krombholz, and E. Weippl. Hardware malware. *Synthesis Lectures on Information Security, Privacy, and Trust*, 2013.

[54] K. Krippendorff. *Content Analysis: An Introduction to It's Methodology*. 2004. ISBN 978-0761915454.

[55] J. S. Laguilles, E. A. Williams, and D. B. Saunders. Can lottery incentives boost web survey response rates? Findings from four experiments. *Research in Higher Education*, 52(5), 2011.

[56] J. Lazar, J. H. Feng, and H. Hochheiser. *Research methods in human-computer interaction*. 2017. ISBN 978-0128053904.

[57] Ledger. A Closer Look Into Ledger Security: the Root of Trust, 2019. https://www.ledger.com/a-closer-look-into-ledger-security-the-root-of-trust/, accessed: 2020-06-17.

[58] Ledger. Export your accounts, 2019. https://support.ledger.com/hc/articles/115005297709, accessed: 2020-06-17.

[59] Ledger. Check hardware integrity, 2020. https://support.ledger.com/hc/articles/115005321449, accessed: 2020-10-08.

[60] M. Leibowitz and J. FitzPatrick. Secure Tokin' & Doobiekeys: How to roll your own counterfeit hardware security devices. DEF CON 25, 2017.

[61] R. V. Lenth. Some practical guidelines for effective sample size determination. *The American Statistician*, 55(3), 2001.

[62] J. Li and J. Lach. At-speed delay characterization for ic authentication and trojan horse detection. In *IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 8–14, 2008.

[63] Y. Li, J. M. McCune, and A. Perrig. VIPER: Verifying the Integrity of Peripherals' Firmware. In *ACM SIGSAC Conference on Computer and Communications Security*, 2011.

[64] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, et al. Meltdown: Reading kernel memory from user space. In *USENIX Security Symposium*, 2018.

[65] M. W. Lipsey. *Design sensitivity: Statistical power for experimental research*. 1989. ISBN 978-0803930636.

[66] P. Maene, J. Götzfried, R. De Clercq, T. Müller, F. Freiling, and I. Verbauwhede. Hardware-based trusted computing architectures for isolation and attestation. *IEEE Transactions on Computers*, 67(3), 2018.

[67] A. Mai, K. Pfeffer, M. Gusenbauer, E. Weippl, and K. Krombholz. User mental models of cryptocurrency systems-a grounded theory approach. 2020.

[68] H. Martin, P. Peris-Lopez, J. E. Tapiador, E. San Millan, and N. Sklavos. Hardware trojans in TRNGs. Citeseer, 2015.

[69] N. R. Mathiasen and S. Bødker. Threats or threads: from usable security to secure experience? In *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*, 2008.

[70] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig. TrustVisor: Efficient TCB reduction and attestation. In *IEEE Symposium on Security and Privacy*, 2010.

[71] Michael Allen. How to Weaponize the Yubikey, 2019. https://www.blackhillsinfosec.com/how-to-weaponize-the-yubikey/, accessed: 2020-09-22.

[72] Mordor Intelligence. Global Hardware Wallet Market: Growth, Trends and Forecast (2019–2024), 2018. https://www.mordorintelligence.com/industry-reports/hardware-wallet-market, accessed: 2020-06-17.

[73] Mordor Intelligence. Hardware OTP Token Authentication Market (2019 - 2024), 2018. https://mordorintelligence.com/industry-reports/hardware-otp-token-authentication, accessed: 2020-06-17.

[74] P. Muir. Trezor hardware wallet is vulnerable to hacking , 2020. https://asiatimes.com/2020/02/trezor-hardware-wallet-is-vulnerable-to-hacking/, accessed: 2020-06-12.

[75] D. Nedospasov, J. Datko, and T. Roth. Wallet Fail, 2018. https://wallet.fail/, accessed: 2020-06-17.

[76] Netflix. Linux and FreeBSD Kernel: Multiple TCP-based remote denial of service vulnerabilities, 2019. https://github.com/Netflix/security-bulletins/blob/master/advisories/third-party/2019-001.md, accessed: 2020-06-17.

[77] N. I. of Standards and Technology. Security Requirements for Cryptographic Modules. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf, accessed: 2020-06-07.

[78] D. Oswald, B. Richter, and C. Paar. Side-channel attacks on the yubikey 2 one-time password generator. In *International Workshop on Recent Advances in Intrusion Detection*, 2013.

[79] C. O'Flynn and A. Dewar. On-Device Power Analysis Across Hardware Security Domains. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019.

[80] B. Parno, J. M. McCune, and A. Perrig. Bootstrapping trust in commodity computers. In *IEEE Symposium on Security and Privacy*, 2010.

[81] J. Payne, G. Jenkinson, F. Stajano, M. A. Sasse, and M. Spencer. Responsibility and tangible security: Towards a theory of user acceptance of security tokens. *preprint arXiv:1605.03478*, 2016.

[82] J.-M. Picod, R. Audebert, S. Blumenstein, and E. Bursztein. Attacking encrypted USB keys the hard (ware) way. *Black Hat USA*, 2017.

[83] P. G. Polson, C. Lewis, J. Rieman, and C. Wharton. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, 36(5), 1992.

[84] P. Qiu, D. Wang, Y. Lyu, and G. Qu. VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies. In *ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[85] M. K. Qureshi. CEASER: Mitigating conflict-based cache attacks via encrypted-address and remapping. In *Annual IEEE/ACM International Symposium on Microarchitecture*, 2018.

[86] S. Rashid. Breaking the Ledger Security Model, 2018. https://saleemrashid.com/2018/03/20/breaking-ledger-security-model/, accessed: 2020-06-17.

[87] J. Reynolds, N. Samarin, J. Barnes, T. Judd, J. Mason, M. Bailey, and S. Egelman. Empirical measurement of systemic 2fa usability. In *USENIX Security Symposium*, 2020.

[88] J. Reynolds, T. Smith, K. Reese, L. Dickinson, S. Ruoti, and K. Seamons. A tale of two studies: The best and worst of Yubikey usability. In *IEEE Symposium on Security and Privacy*, 2018.

[89] F. Richter. The Tech World Is Still a Man's World, 2019. https://www.statista.com/chart/4467/female-employees-at-tech-companies/, accessed: 2020-06-17.

[90] J. Rode, C. Johansson, P. DiGioia, R. S. Filho, K. Nies, D. H. Nguyen, J. Ren, P. Dourish, and D. Redmiles. Seeing further: extending visualization as a basis for usable security. In *Proceedings of the second symposium on Usable privacy and security*, 2006.

[91] G. S. Rose, N. McDonald, L.-K. Yan, and B. Wysocki. A write-time based memristive puf for hardware security applications. In *IEEE/ACM International Conference on Computer-Aided Design*, 2013.

[92] K. Ryan. Hardware-Backed Heist: Extracting ECDSA Keys from Qualcomm's TrustZone. In *ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[93] A. Sabev. Pros and Cons of Secure Elements, 2017. https://www.intrinsic-id.com/pros-cons-secure-elements/, accessed: 2020-06-17.

[94] SatoshiLabs. Is "Banking-grade Security" Good Enough for Your Bitcoins?, 2016. https://blog.trezor.io/284065561e9b, accessed: 2020-06-17.

[95] SatoshiLabs. Non-genuine Trezor One devices spotted. Be careful, buy only from Trezor Shop or authorized resellers, 2018. https://blog.trezor.io/979b64e359a7, accessed: 2020-06-17.

[96] B. Schneier. *Beyond Fear: Thinking Sensibly About Security in an Uncertain World*. 2003. ISBN 978-0387026206.

[97] B. Schneier. Yubico Security Keys with a Crypto Flaw, 2014. https://www.schneier.com/blog/archives/2019/07/yubico_security.html, accessed: 2020-06-10.

[98] K. Sedgwick. Man's Life Savings Stolen from Hardware Wallet Supplied by a Reseller, 2018. https://news.bitcoin.com/mans-life-savings-stolen-from-hardware-wallet-supplied-by-a-reseller/, accessed: 2020-06-17.

[99] A. Seshadri, A. Perrig, L. Van Doorn, and P. Khosla. SWATT: Software-based attestation for embedded devices. In *IEEE Symposium on Security and Privacy*, 2004.

[100] D. W. Stewart and I. M. Martin. Intended and unintended consequences of warning messages: A review and synthesis of empirical research. *Journal of Public Policy & Marketing*, 13(1):1–19, 1994.

[101] R. Strackx and F. Piessens. Fides: Selectively hardening software application components against kernel-level or process-level malware. In *ACM conference on Computer and Communications Security (CCS)*, 2012.

[102] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of ssl warning effectiveness. In *USENIX Security Symposium*, pages 399–416, 2009.

[103] SurveyMonkey. SurveyMonkey, 2019. https://www.surveymonkey.com, accessed: 2020-06-17.

[104] J. Tan, L. Bauer, J. Bonneau, L. F. Cranor, J. Thomas, and B. Ur. Can Unicorns Help Users Compare Crypto Key Fingerprints? In *ACM Conference on Human Factors in Computing Systems*, pages 3787–3798, 2017.

[105] Thetis. U2F Keys, 2020. https://thetis.io/collections/frontpage, accessed: 2020-09-16.

[106] J. Tian, N. Scaife, D. Kumar, M. Bailey, A. Bates, and K. Butler. SoK:" Plug & Pray" Today–Understanding USB Insecurity in Versions 1 Through C. In *IEEE Symposium on Security and Privacy*, 2018.

[107] J. Van Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx. Foreshadow: Extracting the keys to the intel SGX kingdom with transient out-of-order execution. In *USENIX Security Symposium*, 2018.

[108] M. Vervier and M. Orrù. Oh No, Where's FIDO? - A Journey into Novel Web-Technology and U2F Exploitation, 2018. https://www.offensivecon.org/speakers/2018/markus-and-michele.html, accessed: 2020-06-17.

[109] S. Volokitin. Software attacks on hardware wallets. *Blackhat USA*, 2018. https://www.blackhat.com/us-18/briefings/schedule/#software-attacks-on-hardware-wallets-10665.

[110] W3C Fido Alliance. FIDO Alliance and W3C Achieve Major Standards Milestone in Global Effort Towards Simpler, Stronger Authentication on the Web, 2018. https://fidoalliance.org/fido-alliance-and-w3c-achieve-major-standards-milestone-in-global-effort-towards-simpler-stronger-authentication-on-the-web/, accessed: 2020-06-17.

[111] T. Wiki. Developers guide: Cryptography, 2018. https://wiki.trezor.io/Developers_guide:Cryptography, accessed: 2020-06-17.

[112] M. S. Wogalter. Purposes and scope of warnings. *Handbook of warnings*, pages 3–9, 2006.

[113] Yubico. Security Advisory 2019-06-13 – Reduced initial randomness on FIPS keys. https://www.yubico.com/support/security-advisories/ysa-2019-02/, accessed: 2020-06-17.

[114] Yubico. The YubiKey as a Keyboard, 2013. https://support.yubico.com/hc/articles/360013790279, accessed: 2020-12-17.

[115] Yubico. Verify your YubiKey, 2019. https://www.yubico.com/genuine/, accessed: 2020-06-17.

[116] S. Zander, G. Armitage, and P. Branch. A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials*, 9(3), 2007.

[117] S. Zeitouni, D. Gens, and A.-R. Sadeghi. It's hammer time: how to attack (rowhammer-based) DRAM-PUFs. In *Design Automation Conference (DAC)*, 2018.

A Appendix

A.1 Features of other U2F Vendors

Table 6: U2F Token Feature Overview

● fulfilled/implemented/included
◑ sometimes
○ not fullfilled
– not applicable
? undisclosed

		YubiKey 5	YubiKey 4 Neo	YubiKey 4	Yubico Sec. Key	Google Titan	Thetis BLE U2F	Thetis FIDO U2F	Feititan ePass FIDO
Pack.	Tamper-evident	●	◑[1]	◑[1]	◑[1]	○	○	○	◑[1]
	Holographic sticker	○	◑[1]	◑[1]	◑[1]	◑[1]	○	○	◑[1]
Hardware	Single-piece cast	●	●	●	●	●	●	●	●
	Openable device	–[2]	–[2]	–[2]	–[2]	–[2]	–[2]	–[2]	–[2]
	Secure CPU	●	○	○	○	○	○	○	○
	Secure element	○	●	●	●	●	●	●	●
Software	Local FW attestation	?	?	?	?	?	?	?	?
	Remote FW attestation	○	○	○	○	?	?	?	?
	Key attestation	●	●	●	○	?	?	?	?
	Manual firmware load	○	○	○	○	○	○	○	○

[1] Vendors changed their packaging multiple times in recent years.
[2] Single-piece casts and openable devices are mutually exclusive.

A.2 Demographics

Table 7: Demographics of Participants $n = 194$

Demographics	**Participants**	**%**
Gender		
Male	165	85%
Female	15	8%
Other	4	2%
Prefer not to say	10	5%
Age		
<18	5	3%
18–29	94	48%
30–44	81	42%
45–59	12	6%
60+	2	1%
Highest completed education		
Compulsory school	11	6%
Secondary education	49	25%
Bachelor	63	32%
Master	48	25%
PhD	17	9%
Other	6	3%
Continent of residence		
Asia	5	3%
Australia	3	2%
Europe	135	70%
America	50	26%
Prefer not to say	1	1

A.3 Perceived Likelihood of Attack Vectors

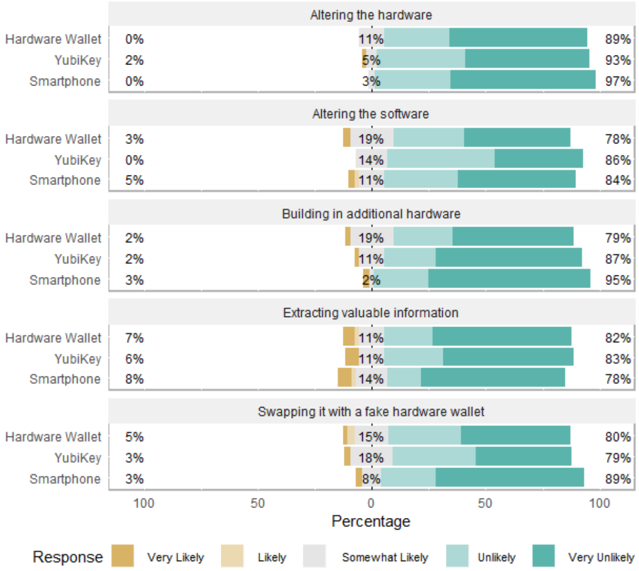

Figure 3: How likely participants perceive attack vectors